

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Алгоритмы и структуры данных»
Тема: Рекурсия

Студент гр. 8304

Ивченко А.А.

Преподаватель

Фирсов М.А.

Санкт-Петербург

2019

Цель работы.

Изучить основы рекурсивного программирования на языке C/C++.

Задание(вариант 7).

Построить синтаксический анализатор для понятия вещественное число.

```
вещественное_число ::= целое_число . целое_без_знака |  
целое_число . целое_без_знака E целое_число |  
целое_число E целое_число  
целое_без_знака ::= цифра | цифра целое_без_знака  
целое_число ::= целое_без_знака | + целое_без_знака | - целое_без_знака
```

Выполнение работы.

В главной функции реализовано два варианта считывания данных: вручную через консоль либо из файла с указанием адреса. Были созданы флаги, которые выполняют функцию регуляторов нецифровых символов в записи вещественного числа.

Функция *IsFloat* обращается к *IsInteger* в зависимости от условий и значений флагов. Так, сперва проверяется, является ли первый символ знаком или цифрой — если да, то значение первого флага меняется, что открывает доступ к обработке символа точки в записи вещественного числа. *IsInteger* рекурсивно обрабатывает цифры, пока не дойдет до точки и обратится к функции *IsFloat*, которая в свою очередь поменяет флаги, тем самым запретив программе обрабатывать еще какие-либо символы точки и сделав «легитимным» символ «E». По достижении символа «E» программа снова сможет один раз пропустить знаки операций(+ и -) и обратится к функции *IsInteger*, которая досчитает цифры до конца строки.

Таким образом, глубина функции *IsInteger* соответствует количеству символов в вещественном числе (если учитывать вызовы функции *IsFloat*).

Краткое описание функций.

- *bool **IsInteger** (std::string str, bool* b, int *flag1, int *flag2)* - рекурсивная функция, глубина которой определяется количеством цифр в числе. Вызывает сама себя, пока не определит символ, отличный от цифры. В этом случае обращается к функции IsFloat.
- *void* **IsFloat** (std::string str, bool* b, int *flag1, int* flag2)* - функция регулирует постановку знаков в вещественном числе. Запускает рекурсию IsInteger в зависимости от правильности расстановки знаков в записи, заданной пользователем.

Тестирование.

	Число	Результат
1.	+34423.423424	True
2.	4.324	True
3.	234.42342E+43	True
4.	5	False
5.	-.2234	False
6.	523467E+234324.234	False
7.	23.23424E+234a	False
8.	3242.00001	False
9.	+234.234+4324E-23	False
10.	234.234342.432424	False

Вывод.

В результате были изучены основы рекурсии на примере работы синтаксического анализатора, определяющего правильность написания вещественного числа.

ИСХОДНЫЙ КОД

```
#INCLUDE <Iostream>
#include <string>
#include <cctype>
#include <fstream>

using namespace std;

//ВЕЩЕСТВЕННОЕ_ЧИСЛО:: = ЦЕЛОЕ_ЧИСЛО.ЦЕЛОЕ_БЕЗ_ЗНАКА |
ЦЕЛОЕ_ЧИСЛО.ЦЕЛОЕ_БЕЗ_ЗНАКАЕЦЕЛОЕ ЧИСЛО |ЦЕЛОЕ_ЧИСЛОЦЕЛОЕ_ЧИСЛО
//ЦЕЛОЕ_БЕЗ_ЗНАКА:: = ЦИФРА | ЦИФРА ЦЕЛОЕ_БЕЗ_ЗНАКА
//ЦЕЛОЕ_ЧИСЛО::=ЦЕЛОЕ_БЕЗ_ЗНАКА | + ЦЕЛОЕ_БЕЗ_ЗНАКА | -ЦЕЛОЕ_БЕЗ_ЗНАКА

void isfloat(string str, bool* b, int* flag1, int* flag2);

bool isinteger(string str, bool* b, int* flag1, int* flag2) { //РЕКУРСИВНАЯ ФУНКЦИЯ,
ГЛУБИНА КОТОРОЙ ОПРЕДЕЛЯЕТСЯ КОЛИЧЕСТВОМ ЦИФР В ЧИСЛЕ

    if (!isdigit(str[0])) {
        if ((str[0] == '.') || (str[0] == 'E') || (str[0] == '+') || (str[0] == '-')) {
            isfloat(str, b, flag1, flag2);
        }
        else {
            *b = false;
            return 0;
        }
    }
    else if (str.length() != 1) {
        isinteger(str.substr(1, 100), b, flag1, flag2);
        return 1;
    }
}
```

```
VOID ISFLOAT(STRING STR, BOOL* B, INT* FLAG1, INT *FLAG2) {//ФУНКЦИЯ РЕГУЛИРУЕТ ПОСТАНОВКУ ЗНАКОВ В ВЕЩЕСТВЕННОМ ЧИСЛЕ
```

```
    IF ((*FLAG1) &&(!*FLAG2)&& (((STR[0] == '+') || (STR[0] == '-')) && ((ISDIGIT(STR[1])) || (ISDIGIT(STR[0]))))) {  
        *FLAG1 = 1;  
        IF (!ISINTEGER(STR.SUBSTR(1, 100), B, FLAG1, FLAG2)) RETURN;  
    }  
    ELSE IF (*FLAG1 && (STR[0] == '.')) {  
        *FLAG1 = 0;  
        *FLAG2 = 1;  
        *B = TRUE;  
        IF (!ISINTEGER(STR.SUBSTR(1, 100), B, FLAG1, FLAG2)) RETURN;  
    }  
  
    ELSE IF ((*FLAG2) && (STR[0] == 'E')) {  
        *FLAG2 = 0;  
        *B = TRUE;  
        IF (!ISINTEGER(STR.SUBSTR(1, 100), B, FLAG1, FLAG2)) RETURN;  
    }  
  
    ELSE {  
        *B = FALSE;  
        RETURN;  
    }  
  
}
```

```
INT MAIN()  
{  
    SETLOCALE(LC_ALL, "RUSSIAN");  
  
    BOOL B = FALSE;  
    INT FLAG1 = 0;// ФЛАГ, ДАЕТ ДОСТУП К СИМВОЛУ '.'  
    INT FLAG2 = 0;// ФЛАГ, ДАЕТ ДОСТУП К СИМВОЛУ 'E'
```

```

COUT << "СЧИТЫВАНИЕ С КОНСОЛИ - 0, С ФАЙЛА - 1\n" << ENDL;
INT A;
CIN >> A;
IF (A){
    STRING FILENAME;
    COUT << "ВВЕДИТЕ ПУТЬ К ФАЙЛУ, КОТОРЫЙ ХОТИТЕ ОТКРЫТЬ:";
    CIN >> FILENAME;
    IFSTREAM FILE(FILENAME);
    IF(!FILE.IS_OPEN()){
        COUT<<"НЕВЕРНЫЙ ПУТЬ К ФАЙЛУ\n";
        RETURN 0;
    }
    STRING STR;

    WHILE (GETLINE(FILE,STR)){

        ISFLOAT(STR, &B, &FLAG1, &FLAG2);//ВЫЗОВ ФУНКЦИИ ISFLOAT
        B ? COUT << STR << " : TRUE\n" : COUT << STR << " : FALSE\n";
        B = FALSE;
        FLAG1, FLAG2 = 0;
    }
    FILE.CLOSE();
}ELSE IF (!A){
    STRING STR1;
    GETLINE(CIN, STR1);
    ISFLOAT(STR1, &B, &FLAG1, &FLAG2);
    B ? COUT << ":TRUE\n" : COUT << ":FALSE\n";

    }RETURN 0;
}

```